Des données aux agents : la simulation réaliste de populations diversifiées de clients

Philippe Mathieu, Sébastien Picault

Des données aux agents : la simulation réaliste de populations diversifiées de clients

P. Mathieu philippe.mathieu@lifl.fr
S. Picault sebastien.picault@lifl.fr

Laboratoire d'Informatique Fondamentale de Lille (UMR CNRS 8022),
Université Lille 1, Cité Scientifique, 59655 Villeneuve d'Ascq, France

Résumé

L'usage croissant de la simulation multi-agents pour modéliser des systèmes pourvoyeurs de grandes quantités de données, suppose l'identification automatique des paramètres pertinents ou l'extraction de connaissances à partir des données réelles, faute de quoi la fiabilité des prédictions et des explications fournies par la simulation est sujette à caution. Dans cet article, nous proposons une méthode pour extraire automatiquement des profils comportementaux à partir de mesures statistiques, dans le cadre de comportements de consommateurs dans un magasin. Dotés des mêmes capacités globales d'interaction, les agents sont munis de profils différents issus de l'exploration des données. Placés dans un magasin virtuel réaliste, dans lequel tous leurs objectifs peuvent ne pas être atteignables, ils effectuent néanmoins des achats qui reflètent la diversité des clients réels ainsi que les profils initiaux. Nous défendons l'idée que de telles techniques sont nécessaires pour faire des simulations multi-agents un puissant outil d'aide à la décision.

Mots-clés : *Simulation multi-agents, Exploration de données, Marketing, Interactions*

Abstract

The growing use of multiagent-based simulation for modeling systems associated with very large databases, addresses specific issues such as the automatization of parameter identification or knowledge extraction from real data, so as to enhance the confidence in simulation predictions and explanations. In this paper, we propose a method for automatically retrieving behavioral prototypes from statistical measures, in the context of consumer behavior. Endowed with the same overall behavior, the agents are given different profiles based on the data analysis. They are put into a spatially realistic store, where some of their objectives may be unattainable. Though, their purchase reproduce the original clusters. We argue that such techniques are essential to make multi-agent simulations a powerful decision support tool.

Keywords: *Agent-based simulations, Knowledge discovery, Marketing, Interactions*

1 Introduction

Depuis de nombreuses années déjà, les modèles centrés individus et les simulations multi-agents sont employés pour renforcer la compréhension de systèmes complexes large échelle, et ce dans des domaines variés, de la biologie moléculaire aux réseaux sociaux. Ces approches éclairent en effet les mécanismes qui font émerger des phénomènes collectifs à partir des interactions entre les entités du système, en plus de fournir des prédictions sur des variables macroscopiques. Or, les domaines récemment abordés par la simulation multi-agents sont de ceux qui produisent d'énormes quantités de données. Parmi les nouvelles problématiques qui en découlent, se pose de façon sensible la question de l'intégration de connaissances construites automatiquement à partir de ces données, en vue de compléter voire remplacer une expertise humaine.

L'approche que nous proposons ici tente d'extraire autant d'information que possible de données enregistrées afin d'identifier des groupes d'agents dont les traces d'activité sont similaires. Nous construisons une description abstraite de leurs buts (sous forme de prototypes), laquelle permet de simuler une population d'agents qui reflète la diversité originelle. Nous adaptons pour ce faire diverses techniques de fouille de données au contexte particulier de l'initialisation de sous-groupes d'une population d'agents. Nous montrons que les prototypes ainsi construits, combinés à un modèle de comportement approprié et à une action située, produisent des traces d'activité statistiquement réalistes.

L'article est structuré comme suit : la section 2 présente le cadre de nos travaux, notamment le contexte de l'analyse et de la simulation de com-

portements de clients, qui nous a servi de cadre applicatif. La section 3 décrit la manière dont nous représentons les informations pertinentes pour l'identification et la caractérisation des articles d'un magasin, des transactions effectuées (achats) et des prototypes qui peuvent en être inférés. La section 4 présente le processus d'exploration de données proprement dit, i.e. comment construire des prototypes à partir des transactions ; et la section 5, comment nous avons évalué sa robustesse et mis en œuvre notre approche au sein d'une simulation multi-agents.

2 Contexte scientifique

La question générale de l'identification statistique des caractéristiques d'une population d'agents à partir de données se pose avec une acuité croissante, comme en témoignent d'ailleurs plusieurs travaux récents, tels que la détection dynamique de groupes émergents [4] ou encore l'intégration dans les simulations de paramètres appris afin d'exprimer une diversité comportementale [11]. Dans cet article, nous nous plaçons dans la situation où les traces pertinentes de l'activité des agents sont représentables sous formes de *transactions* au sens d'Agrawal [1]. Il s'agit en l'occurrence du contexte applicatif de simulation de comportements de clients dans un magasin, où les transactions enregistrées sont des *tickets de caisse*, i.e. un ensemble d'articles ; nous discutons ultérieurement de la généralisation de notre approche à d'autres domaines. Nous commençons par exposer diverses approches utilisées pour l'analyse des paniers d'achats et la segmentation clientèle, puis nous décrivons le modèle multi-agents sur lequel nous nous appuyons pour tester notre approche.

2.1 Les techniques d'analyse et de simulation des comportements de clients

Les techniques classiques utilisées en marketing, par exemple pour partitionner les consommateurs en sous-groupes ayant des habitudes similaires, ou pour détecter des articles achetés fréquemment ensemble, consistent à extraire des informations globales à partir de très grandes bases au moyen d'algorithmes de fouille de données dédiés.

La principale technique de recherche d'informations dans des données réelles est l'analyse d'affinité [1], qui s'appuie sur la co-ocurrence d'articles dans les achats enregistrés. Cette méthode peut être appliquée directement à des tickets de caisse réels et permet d'inférer des règles d'association entre articles (i.e. $X \rightarrow Y$ où X, Y sont des ensembles disjoints d'articles). Ces règles sont caractérisées par un *support* (proportion des achats qui contiennent à la fois X et Y) et une *confiance* (probabilité conditionnelle d'acheter les articles de Y lorsque ceux de X sont dans le panier).

Cette approche est très efficace pour la vente additionnelle (*cross-selling*) ou la montée en gamme (*up-selling*), et dans une certaine mesure donne des indications pour le placement des produits en rayon (par exemple, mieux vaut essayer d'associer dans les linéaires des produits achetés fréquemment ensemble). La première de ses limitations est le temps de calcul, qui croît comme le cube du nombre d'articles [5]. Mais surtout, il est assez difficile d'utiliser les règles d'association pour diriger les achats des agents, car une règle ne fait que suggérer des produits qui sont *associés à d'autres*, sans indiquer comment amorcer le panier [16].

Une autre méthode consiste à essayer de prédire des *listes de courses* à partir des tickets. Par exemple, dans [7], un assistant personnel tente d'apprendre les habitudes d'achats individuelles afin de rappeler au consommateur ce dont il va le plus certainement avoir besoin lors de ses prochaines courses, et de lui proposer des promotions sur mesure. Le but de cette application est très éloigné du nôtre, et les méthodes de classification employées ne construisent pas de représentation symbolique de la liste de courses : elles ne font qu'une prédiction probabiliste sur des catégories générales de produits. Néanmoins, ce travail démontre la possibilité d'opérer un apprentissage inductif sur des tickets réels dans le but d'identifier une liste de courses sous-jacente.

Quant aux modèles centrés individus, ils sont utilisés de façon croissante depuis quelques années dans le domaine de la distribution pour aider la prise de décision marketing [15, 17, 18, 13], dans la mesure où la modélisation fine des comportements individuels permet d'élucider les raisons de l'efficacité (ou non) d'une technique commerciale donnée. Ces modèles permettent de prendre en compte les préférences au niveau individuel, et même d'introduire une expertise psychologique [19], de façon à construire une description précise des motivations et des besoins de chaque consommateur. Ce sont ensuite les actions de ces clients simulés qui sont responsables des achats pré-

dits. Les hypothèses concernant les facteurs qui influencent les ventes peuvent être décrites explicitement dans le modèle : elles peuvent être comprises et examinées par les experts, et validées ou invalidées au moyen d'expériences appropriées.

En contrepartie, ces modèles multi-agents, tout particulièrement lorsqu'ils mettent en œuvre des agents cognitifs, requièrent souvent une expertise qui n'est facile ni à acquérir ni à implémenter. De plus, peu de modèles de simulation de magasins prennent en compte les aspects spatiaux qui sont pourtant considérés comme essentiels dans la grande distribution, comme l'allocation des linéaires, le placement des articles, le dimensionnement des caisses, la publicité sur le point de vente, etc. Dans de précédents travaux [18], nous avons abordé ces questions de façon à concevoir une simulation de supermarchés où les agents sont situés dans un environnement spatialement réaliste. Ce caractère situé, comme nous le montrons plus loin, est nécessaire pour transformer une simulation *ad hoc* en un véritable outil d'aide à la décision, capable de prédire comment les clients réagissent aux changements de l'organisation spatiale du magasin, aux événements commerciaux ou encore à la pression concurrentielle. D'autre part, la fidélité des profils de consommateurs doit être respectée, afin d'assurer le réalisme non pas uniquement des actions effectuées par les agents simulés, mais également de la diversité de leurs objectifs d'achats.

2.2 Un modèle orienté interactions

L'approche que nous défendons ici s'inscrit dans la continuité de nos travaux antérieurs sur ce sujet [18, 13], dans lesquels l'acquisition d'une expertise et la conception de divers modèles de simulation imposaient une démarche incrémentale et empirique, d'où l'usage de la méthode IODA [10] pour aider les psychologues et les experts marketing d'exprimer explicitement les règles comportementales adoptées par les agents.

Pour mémoire, cette méthode « orientée interactions » considère que toute entité du modèle est représentée par un agent ; chaque comportement est modélisé de façon autonome par une « interaction », i.e. une séquence d'actions impliquant un agent source et un agent cible, soumise à des conditions d'exécution. Une interaction est réalisable si la source et la cible satisfont les conditions. Agents et interactions peuvent être développés en bibliothèques indépendantes, puis la simulation est conçue en assignant des interactions à des couples d'agents, au sein d'une « matrice d'interaction » (cf. tab. 1), qui constitue un moyen très visuel d'exprimer quelles familles d'agents sont autorisées à interagir, et au moyen de quelles interactions. Cette matrice d'interaction est traitée par un moteur de simulation générique, qui a pour fonction principale d'évaluer les interactions réalisables avec leurs sources et cibles respectives, de façon à déterminer quelles actions doivent être effectuées par chacun des agents.

Le modèle de comportement sur lequel nous nous appuyons dans les travaux présentés ici, a été développé pour la simulation d'un magasin de détail dans le but de former des vendeurs en les confrontant à des clients simulés au sein d'un *Serious Game* immersif (projet FormatStore [13]). Le comportement générique des clients simulés a donc été validé par des experts en marketing et, dans toute la suite, il est *considéré comme figé*. La matrice d'interaction correspondante figure dans le tableau 1.

En résumé, les clients ont un même comportement d'ensemble, mais *diffèrent dans leurs besoins*, aussi sont-ils dotés au démarrage de la simulation d'une *liste de courses* qui spécifie, de façon plus ou moins détaillée, quels articles ils sont susceptibles d'acheter dans le magasin. Ces besoins peuvent être spécifiés avec précision (par exemple « SodaCola light, pack de $6\times$ 2L ») ou au moyen d'une description vague (telle que « eau de source ») qui peut s'appliquer à de nombreux articles du magasin. Cette liste de courses est utilisée par les conditions de l'interaction *Take* (qui peut être effectuée par un agent Customer sur un agent Item) pour rendre compte de la décision d'achat.

Au cours de la simulation, les interactions ont lieu selon la matrice d'interaction et l'état et la position des agents, en respectant un comportement cohérent pour les consommateurs : les clients artificiels cherchent tous les articles qui figurent sur leur liste, durant un laps de temps fixé. Ils peuvent obtenir diverses informations utiles pour accomplir cette tâche à partir des affiches, panneaux, caisses, articles, etc.

Dans FormatStore [13], ces listes de courses étaient soit aléatoires (selon une distribution de Poisson basée sur la taille moyenne des paniers), soit définies « à la main » selon une scénarisation particulière destinée à placer l'apprenti vendeur dans une situation probléma-

TABLE 1 – Matrice d'interaction qui définit le comportement de tous les agents de la simulation. Par exemple, l'intersection entre la ligne Customer et la colonne Item contient deux interactions, *Take* et *MoveTowards*, ce qui signifie qu'un agent Client peut soit prendre un agent Article, soit s'en approcher, en fonction de la priorité (premier des deux nombres — ici, *Take* a la plus forte priorité) et de la distance entre les agents (second nombre), sous réserve que les conditions des ces interactions soient satisfaites pour chacun des agents. La colonne $\emptyset$ contient les interactions réflexives (i.e. où l'agent cible est l'agent source lui-même).

Source \ Target	$\emptyset$	Customer	Item	Checkout	Queue	Door
Customer	Wander (0) GoToPlace (1, ∞)		MoveTowards (2, 10) Take (4, 1)	MoveTowards (3, 10)	StepIn (5, 2) MoveOn (6, 1) WalkOut (7, 1)	Exit (8, 1)
Item		Notify (1, 10)				
Sign		Notify (1, 10)				
Checkout	Open (10) Close (10)	Notify (1, 15) CheckOut(7, 1)			Handle (8, 1) ShutDown (9, 1)	
Door	SpawnCustomer(1)	Notify(1, 10)				

tique. Dans les expériences décrites dans la section 5, ces listes ont été construites au moyen de l'algorithme d'extraction de connaissances que nous proposons. La question que nous discutons dans la suite est donc essentiellement : comment construire efficacement de telles listes de courses ?

La réponse triviale consisterait à utiliser telles quelles les transactions enregistrées pour « rejouer » en simulation les achats réels. Mais les achats réels ne sont pas la cause du comportement des clients : seulement leur effet. Ils résultent précisément de la confrontation de deux mécanismes : d'une part, des besoins plus ou moins précis chez les clients, et d'autre part, une certaine organisation spatiale du magasin, la disponibilité ou la visibilité des articles, etc. autrement dit le caractère situé des choix que doivent faire les individus dans leur environnement. Les utiliser en dépit de ce caractère *fortuit* priverait la simulation de toute capacité d'extrapolation.

Nous tentons donc de combiner *l'identification de clients similaires* (comme dans la segmentation clientèle classique) en classifiant leurs transactions, et *l'induction d'une caractérisation abstraite de ces classes*, en l'occurrence par des listes de courses types (prototypes). Nous pouvons dès lors utiliser l'association entre des classes de clients et leurs listes de courses pour générer des profils d'agents susceptibles d'acheter des articles similaires. Notre démarche peut être vue comme une boucle méthodologique comme celle présentée sur la figure 1 : à partir des transactions (tickets de caisse) enregistrés, nous pouvons extraire des prototypes représentatifs de divers groupes de transactions similaires (listes de courses) ; après quoi, ces prototypes peuvent être utilisés en simulation pour produire à leur tour des transactions simulées, qui peuvent être elles-mêmes analysées par le même procédé pour vérifier que les prototypes issus de la simulation sont les mêmes que ceux issus des données réelles.

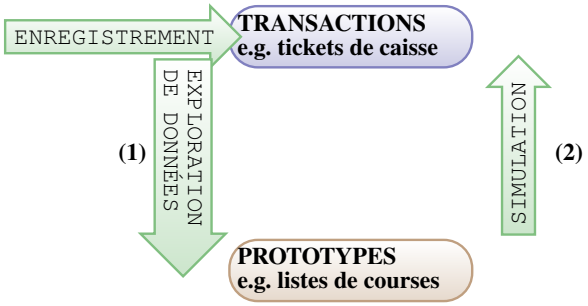


FIGURE 1 – Un aperçu de notre démarche méthodologique. (1) Notre processus d'exploration de données construit automatiquement des prototypes à partir de transactions. (2) La simulation utilise les prototypes pour paramétrer les comportements et produire des transactions simulées, qui à leur tour peuvent être segmentées en prototypes à comparer à ceux issus des données réelles.

Pour rentrer dans les détails de ce processus, nous devons d'abord expliquer comment nous allons décrire les articles, les tickets de caisse et les listes de courses pour leur appliquer le traitement le plus adapté.

3 Représentation des connaissances

3.1 Identifiants des articles

Dans la grande distribution, chaque produit unique est identifié par une « unité de gestion des stocks » (UGS, en anglais SKU pour « stock-keeping unit ») afin de tracer sa disponi-

bilité et la demande. Les UGS ne sont pas destinés à décrire la nature ou les caractéristiques des produits de façon standardisée, mais seulement à les référencer. Ces méthodes d'identification ne sont pas très adaptées pour extraire autre chose de des règles de co-occurrence. Il est en fait nécessaire, pour caractériser ces produits en vue d'une analyse explicative, de leur adjoindre des connaissances marketing pertinentes (famille de produits, qualité, prix relatif, marque, label bio, etc.). Cette étape peut requérir une expertise pour évaluer quelles sont les caractéristiques jugées pertinentes dans un contexte commercial donné.

Dans ce qui suit, nous identifions chaque produit unique par un **tuple d'entiers strictement positifs**, qui encode les valeurs des caractères jugés pertinents. Par exemple, si les propriétés utiles sont la marque, la famille du produit et une description détaillée (e.g. respectivement « Soda-Cola », « boisson », « soda goût cola »), alors les produits sont identifiés par un triplet d'entiers, e.g. (31, 4, 15). Cela permet une représentation de tous les produits à une granularité arbitrairement fine, incluant des qualifications très spécifiques comme « bio », « commerce équitable » ou « sans gluten ». De plus, des valeurs continues comme le prix ou le poids peuvent être encodées moyennant une discrétisation préalable (e.g. 1 pour « bon marché », 2 pour « moyen », 3 pour « cher » ; ou de 1 à 4 pour un conditionnement variant de « petit » à « familial »). La transformation des UGS ou d'autres méthodes d'identification en ce genre de tuples d'entiers peut être effectuée automatiquement, en effectuant les jointures appropriées entre bases de données.

3.2 Transactions et prototypes

Notre processus d'exploration de données s'appuie sur des informations d'achats enregistrées ; les plus accessibles sont les tickets de caisse. Une *transaction* au sens de [1] peut être calculée à partir d'une simple énumération des produits uniques (sans doublons) présents sur les tickets, sans prendre en compte les quantités (comme [1]). Ainsi, à partir d'une liste d'identifiants de type UGS, nous construisons un ensemble de tuples d'entiers.

À partir de ces transactions, le processus que nous appliquons consiste à extraire des *prototypes* qui visent à décrire un « ticket abstrait » caractérisant les groupes de tickets similaires. Pour ce faire, nous introduisons d'abord la notion d'*article prototype*. Un article prototype est également un tuple d'entiers, mais pour lequel **la valeur 0 est autorisée comme valeur générique (joker)**. Par exemple, un produit caractérisé par « n'importe quelle marque », « boisson », « soda goût cola » peut être décrit au moyen du triplet (0, 4, 15). Le triplet nul (0, 0, 0) signifie « n'importe quel produit ». Un *prototype* est alors simplement défini comme un ensemble d'articles prototypes.

Ces prototypes construits à partir des données peuvent également être utilisés comme « listes de courses » pour les cliens simulés, car il arrive fréquemment que seuls quelques traits des articles souhaités soient effectivement spécifiés. Ainsi, M. Dupont achète du « soda goût cola » sans égard pour la marque, tandis que M. Dupond est susceptible d'acheter n'importe quel yaourt bio de la marque « Yoopla ». L'usage du 0 comme joker est fort utile pour exprimer de tels souhaits vagues.

Dans la section suivante, nous montrons comment de tels prototypes sont effectivement construits à partir des transactions.

4 Étapes du processus d'exploration de données

L'analyse des achats procède de la façon suivante : 1° la base de transactions est partitionnée en classes (ce qui suppose de définir préalablement une mesure de distance entre tickets, elle-même basée sur une mesure de distance entre articles) ; 2° pour chaque classe de transactions, tous les articles qui apparaissent sur les transactions sont à leur tour classés pour construire des articles prototypes, puis le prototype composé de l'union des articles prototypes est évalué par rapport à toutes les transactions de la classe.

4.1 Mesure de similarité entre articles

Comme certains articles souhaités par les clients peuvent n'être pas complètement spécifiés, il est à prévoir que des consommateurs dotés du même prototype (i.e. de la même liste de courses) *n'achètent pas exactement les mêmes produits*. Aussi, si l'on calcule une distance entre transactions uniquement en fonction des produits qu'elles ont en commun, il faut s'attendre à ce que la classification des transactions soit de piètre qualité. Nous proposons au

contraire de moduler la comparaison des transactions en prenant en compte la distance entre articles.

Une façon simple de procéder est de calculer une distance « à la Hamming » (ou réciproquement un indice de similarité de type Hamming). Si deux articles (ou articles prototypes) sont représentés par les tuples d'entiers $I = (f_1, ..., f_n)$ et $I' = (f'_1, ..., f'_n)$, leur similarité est définie comme suit : $\sigma(I, I') = \frac{1}{n} \sum_{i=1}^{n} \varsigma(f_i, f'_i)$ où $\varsigma(f_i, f'_i) = 1$ si $f_i = f'_i$ ou $f_i = 0$ ou $f'_i = 0$, et $\varsigma(f_i, f'_i) = 0$ sinon.

4.2 Mesure de similarité entre transactions

Afin de comparer les transactions, nous sommes partis d'une mesure fort employée pour le calcul de similarités entre ensembles de tailles différentes : l'indice de Jaccard [9]. Il est défini comme suit pour tout ensemble X, Y : $J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X|+|Y|-|X \cap Y|}$

Comme nous l'avons expliqué ci-dessus, l'usage brut de l'indice de Jaccard ne peut satisfaire nos besoins, dans la mesure où il ne fait aucune différence entre des ensembles disjoints et des ensembles qui contiennent des éléments proches mais différents. Aussi, nous en proposons une extension, basée sur la similarité entre articles. Elle consiste à calculer le score des meilleurs appariements entre les articles de deux transactions, par similarité décroissante. Ce nouvel indice est noté dans la suite J_{BM} (pour « best-match Jaccard index »). Pour le calculer entre une transaction $\mathcal{T} = \{I_1, ..., I_p\}$ et une autre transaction $\mathcal{T}' = \{I'_1, ..., I'_q\}$, nous appliquons l'algorithme suivant :

1. Calculer la matrice d'appariement $(\sigma_{i,j})$ avec $\sigma_{i,j} = \sigma(I_i, I'_j)$
2. Pour k de 1 à $\min(p, q)$:
   (a) Calculer $\mu_k = \max_{i,j}(\sigma_{i,j})$
   (b) Identifier un couple $(i^\star, j^\star)$ tel que $\sigma_{i^\star,j^\star} = \mu_k$ (si plusieurs couples (i, j) vérifient $\mu_k = \sigma_{i,j}$, sélectionner un couple $(i^\star, j^\star)$ qui minimise : $(\sum_{i \neq i^\star} \sigma_{i,j^\star} + \sum_{j \neq j^\star} \sigma_{i^\star,j})$)
   (c) Remplacer $(\sigma_{i,j})$ par la sous-matrice obtenue en supprimant la ligne $i^\star$ et la colonne $j^\star$
3. $\mu_{BM} = \sum_{k=1}^{\min(p,q)} \mu_k$ joue le même rôle que $|X \cap Y|$ dans l'indice de Jaccard classique, de sorte que nous avons :

$$J_{BM}(\mathcal{T}, \mathcal{T}') = \frac{\mu_{BM}}{p + q - \mu_{BM}}$$

Par exemple, prenons $\mathcal{T}$ ={(1, 1, 2), (3, 5, 8), (13, 21, 34)} et $\mathcal{T}'$={(1, 1, 2), (3, 6, 8), (12, 13, 14), (1, 1, 34)}. Comme on a $\mathcal{T} \cap \mathcal{T}'$ ={(1, 1, 2)} seulement, on aurait $J(\mathcal{T}, \mathcal{T}') \approx 0.17$, tandis que l'on obtient $J_{BM}(\mathcal{T}, \mathcal{T}') = 0.4$, car plusieurs articles de $\mathcal{T}$ et de $\mathcal{T}'$ sont assez proches comme on peut le voir sur le tableau 2 ci-après.

TABLE 2 – Exemple de matrice de similarité entre les articles de deux transactions, utilisée pour calculer la valeur de J_{BM}. Les valeurs des μ_k sont en gras. On a ici $\mu_{BM} = 2$ d'où $J_{BM} = \frac{2}{3+4-2} = 0.4$

$\sigma(I, I')$	(1, 1, 2)	(3, 6, 8)	(12, 13, 14)	(1, 1, 34)
(1, 1, 2)	**1**	0	0	0.6666667
(3, 5, 8)	0	**0.6666667**	0	0
(13, 21, 34)	0	0	0	**0.3333333**

Il existe de nombreuses mesures de similarité et de distance qui peuvent se substituer à l'indice de Jaccard [6] ; en pratique, la méthode que nous proposons est également adaptée pour les plus fréquents d'entre eux tels que les indices d'Ochiai [14] ou de Sørensen-Dice [8], qui peuvent être étendus suivant le même algorithme de meilleur appariement. Ce point précis a été vérifié expérimentalement au moyen de la même procédure que celle que nous présentons dans la section 5.

4.3 Classification des transactions

Nous utilisons ensuite l'indice J_{BM} pour calculer une matrice des distances entre toutes les transactions de la base de données : $\Delta_{BM} = (\mathrm{d}_{i,j})$ avec $\mathrm{d}_{i,j} = 1 - J_{BM}(\mathcal{T}_i, \mathcal{T}_j)$. Cette matrice de distances peut être soumise à de nombreuses techniques de classification. Nous avons choisi un algorithme très classique de classification hiérarchique, en l'occurrence celui implémenté dans la bibliothèque `flashClust` de R [12], et qui nous permet d'obtenir très simplement des dendrogrammes en fonction des similarités entre transactions.

Ensuite, le dendrogramme est découpé en K classes (e.g. au moyen de la fonction `cutree` de R). La valeur « correcte » de K n'est pas aisée à déterminer *a priori*, aussi nous avons tenter d'évaluer empiriquement la hauteur la plus appropriée pour couper l'arbre, au moyen de

prototypes générés aléatoirement avec une valeur connue pour K (cf. section 5). Il est apparu qu'une hauteur $h \approx 0.6 - 0.7$ permettait dans tous les cas de trouver la bonne valeur de K.

4.4 Induction des prototypes

Construire un prototype pour une classe consiste ici à trouver un ensemble de tuples (autorisant le 0) qui obtient le meilleur score possible, quand on mesure sa similarité avec les N transactions de la classe au moyen de l'indice J_{BM}. Il doit donc être aussi précis que possible tout en introduisant de la généralisation lorsque c'est nécessaire.

Le processus commence par une analyse de fréquence : pour chaque article I qui apparaît sur l'une des N transactions de la classe à analyser, nous calculons $\phi(I)$ comme le rapport entre le nombre de transactions contenant I, et N.

Les articles *rares*, i.e. pour lesquels $\phi(I) < \varepsilon$, peuvent être considérés comme des achats de circonstance ou du « bruit », et simplement ignorés (en pratique cela fonctionne assez bien pour $\varepsilon \approx \frac{1}{N}$, i.e. des articles qui n'apparaissent que sur un seul ticket). Inversement, les articles *fréquents*, i.e. pour lesquels $\phi(I) > \theta$, peuvent être considérés comme indispensables et sont maintenus tels quels dans le prototype à construire (empiriquement, nous utilisons $\theta \approx 0.95$).

En ce qui concerne les articles de fréquence intermédiaire, nous leur appliquons une nouvelle classification, afin de détecter certaines régularités, par exemple que les produits « SodaCola » sont toujours associés à du « yoghourt bio » de diverses marques. Pour cela nous calculons une matrice des distances entre articles de la classe considérée : $(D_{i,j})$ avec $D_{i,j} = 1 - \sigma(I_i, I_j)$, et nous l'utilisons pour construire un dendrogramme des articles. À nouveau, le nombre de classes pertinentes K_I n'est pas connu *a priori*.

Pour l'estimer, nous utilisons l'algorithme suivant pour les valeurs possibles de K_I :

1. Pour chaque classe d'articles : calculer un article prototype en plaçant un 0 lorsque les caractéristiques diffèrent ; par exemple, si les articles de la classe sont (1, 5, 7), (1, 6, 7) and (1, 12, 7), l'article prototype correspondant est (1, 0, 7).
2. Faire l'union de tous les articles prototypes ainsi que des articles fréquents, ce qui donne un prototype candidat $\mathcal{P}_{K_I}$.
3. Calculer le score de $\mathcal{P}_{K_I}$ comme la moyenne de $J_{BM}(\mathcal{P}_{K_I}, \mathcal{T})$ pour toutes les transactions $\mathcal{T}$ de la classe analysée.

Nous conservons la valeur $K_I^\star$ pour laquelle le prototype associé $\mathcal{P}_{K_I^\star}$) maximise ce score. Cet algorithme est appliqué aux K classes de transactions pour générer K prototypes.

5 Validation

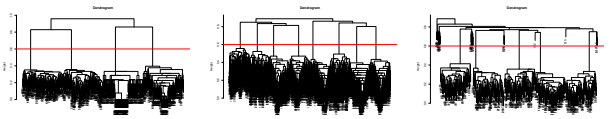
Les prototypes ainsi obtenus ont vocation à être utilisés par un processus de simulation pour produire de transactions artificielles. Les agents clients effectuent des comportements de façon autonome, en fonction de leurs listes de courses contenant des articles prototypes (i.e. avec des jokers), et ce dans le contexte situé d'un magasin réaliste. Dans la mesure où des articles peuvent manquer ou être difficiles à trouver, on ne peut toutefois s'attendre à ce que les transactions qui résultent de la simulation soient exactement identiques aux transactions réelles qui sont à l'origine des listes de courses.

Pourtant, nous devons nous assurer que la simulation est capable de reproduire le même *type de comportement d'achat* que celui observé chez les clients réels. Un moyen d'y parvenir est d'analyser les transactions produites par la simulation, de reconstruire les prototypes correspondants au moyen du même processus de fouille de données, et de les comparer aux prototypes issus des données réelles.

Toutefois, il est nécessaire avant d'analyser les résultats de simulations multi-agents de vérifier que la méthode de construction des prototypes est suffisamment robuste. Sans cela, d'éventuelles différences entre les prototypes qui reflètent l'activité des agents et ceux qui caractérisent les achats de consommateurs réels, pourraient trouver leur origine non pas dans le comportement des agents ou dans les particularités de l'environnement, mais seulement dans une forte sensibilité aux perturbations du processus d'exploration des données. Nous présentons donc ci-dessous comment la robustesse de notre méthode d'analyse a été évaluée.

5.1 Simulations stochastiques de l'instanciation des prototypes

Afin d'effectuer des tests assez complets, nous avons généré plusieurs ensembles de prototypes, chacun composé d'articles prototypes aléatoires. Puis, pour obtenir une simulation à



(a) (b) (c)

FIGURE 2 – Dendrogrammes issus des similarités entre transactions dans 3 expériences. La ligne horizontale représente la hauteur de coupe (0.6) qui permet de retrouver les K classes de départ. Paramètres : (a) $K = 4, N_I = \{5, 10, 20, 40\}, N_T = 200, N_A = 5\ \%, N_M = 5\ \%, N_O = 0$; (b) $K = 8, N_I = 10, N_T = 400, N_A = 5\ \%, N_M = 5\ \%, N_O = 0$; (c)$K = 5, N_I = 20, N_T = 100, N_A = 5\ \%, N_M = 5\ \%, N_O = 5\ \%$.

gros grain, mais rapide, des achats induits par ces prototypes, nous les avons instanciés de façon aléatoire, sur la base des paramètres suivants :

- le nombre K de classes (donc de prototypes) à tester ;
- le nombre d'articles prototypes $N_I(i)$ dans chaque classe i ;
- le nombre de transactions par classe, $N_T(i)$, qui détermine combien de transactions sont instanciées pour le prototype i ;
- le nombre d'articles additionnels $N_A(i)$ qui indique combien d'articles aléatoires sont ajoutés dans chaque transaction de la classe i (cela permet de représenter des achats occasionnels qui ne sont pas représentatifs des habitudes des agents de cette classe) ;
- le nombre d'articles manquants $N_M(i)$ qui donne, pour la classe i, le nombre d'articles prototypes qui ne sont pas instanciés (possibilité que certains articles ne soient pas trouvés) ;
- le nombre N_O de transactions qui n'appartiennent à aucune classe (et sont générées de façon totalement aléatoire).

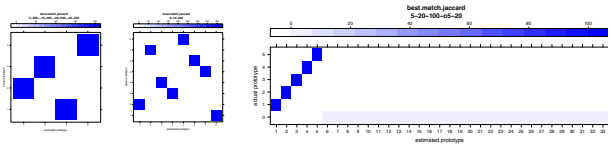
L'*instanciation* d'un article prototype consiste à remplacer chaque 0 par un entier aléatoire strictement positif, dans un certain domaine de valeurs pris par les caractéristiques des articles réels. Dans nos expériences nous avons utilisé des tuples de 5 entiers dont les valeurs maximales étaient (20, 100, 10, 5, 2) (valeurs choisies sur la base de travaux antérieurs [13]).

Nous avons mené des expériences automatiques et leur évaluation pour les combinaisons des paramètres ci-dessus dans les intervalles suivants : K : 4 – 10 ; N_I : 5, 10, 20, 40 ; N_T : 50, 100, 200, 400, 800 ; N_A : 0, 5, 10 % du nombre d'articles dans les transactions ; N_M 0, 5, 10 % du nombre d'articles dans les transactions ; N_O : 0, 5, 10 % du nombre total de transactions. Toutes ces expériences ont donné des résultats concordants que nous résumons ci-après.

5.2 Résultats et discussion

Comme le montre la figure 2 pour trois des ces expériences, les transactions produites par l'instanciation de prototypes aléatoires sont correctement discriminées : couper les arbres à une hauteur voisine de 0.6 suffit à identifier des classes qui reflètent exactement les K classes d'origine (cf. fig. 3a–b), même lorsque les transactions sont construites avec des articles additionnels ou manquants.

Lorsque la base contient également des transactions aléatoires (cf. fig 2c), i.e. n'appartenant à aucune classe caractérisée par un prototype, le processus identifie toujours correctement les K classes d'origine, en ajoutant des classes supplémentaires très petites (cf. fig. 3c), qui sont la plupart du temps réduites à une seule transaction. Ces classes peuvent être très facilement écartées lors de la phase de généralisation. Dans toutes les expériences (jusqu'à $N_O = 10\ \%$ du nombre total de transactions), la construction de prototypes a été un succès, ce qui nous semble une bonne indication de robustesse.



(a) (b) (c)

FIGURE 3 – Comparaison entre les classes de transactions estimées (abscisses) et les classes d'origine (ordonnées) pour les expériences (a), (b) et (c). L'intensité de chaque carré est proportionnelle au nombre de transactions classées dans la ligne et la colonne correspondantes. Tandis que (a) et (b) montrent une correspondance exacte, dans (c) il y a des transactions « bruit » générées de façon totalement aléatoire : elles ne sont *pas* réparties parmi les véritables classes, mais placées dans de petits groupes isolés.

5.3 Expérimentations multi-agents

Le simulateur conçu dans nos travaux antérieurs (cf. [13] et § 2.2) a été modifié pour représenter les listes de courses au moyen de prototypes et les articles par des tuples d'entiers.

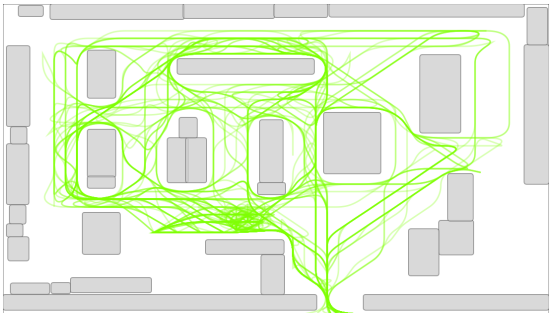


FIGURE 4 – Tracé des chemins suivis par les clients simulés dans le magasin virtuel.

Nous avons d'abord mené des expériences avec les mêmes prototypes que dans les simulations stochastiques et avec des agents connaissant la position des rayons dans le magasin. Dans ces conditions, les transactions simulées sont capables de reproduire les classes et les prototypes de départ.

Cependant, ces résultats sont modulés par la *limite temporelle* donnée aux agents pour effectuer leurs achats, lorsque ceux-ci ne connaissent pas l'agencement spatial du magasin. Lorsqu'elle est trop réduite, ils sortent du magasin sans avoir acheté tous les articles de leur liste. Cela n'a pas d'effet sur l'identification des classes de transactions (grâce à la robustesse du processus vis-à-vis des articles manquants), mais peut modifier la nature des prototypes construits à partir des transactions simulées. En effet, l'observation des chemins suivis par les clients dans le magasin (cf. fig. 4) met en évidence des « points chauds » très fréquentés, où les articles sont donc trouvés facilement, ainsi que des « points froids » où les agents passent peu : les produits manquants sont donc souvent les mêmes car « mal placés » (du point de vue marketing), contrairement à ce qui se passait dans les simulations stochastiques où les produits manquants étaient choisis de façon uniformément aléatoire parmi les articles prototypes de la liste de courses. Il en résulte que les prototypes reconstruits dans ces conditions constituent un sous-ensemble des prototypes d'origine.

Loin de constituer une limitation de notre approche, cettre propriété met en lumière le côté crucial de la mise en situation spatiale des simulations, et illustre bien comment utiliser ces outils pour l'aide à la prise de décision quant au placement des articles en rayon, l'agencement du magasin, la signalisation, etc. La limitation du temps passé dans le magasin est également importante, non seulement parce qu'elle serait un gage de réalisme, mais plus significativement parce qu'elle exerce une pression forte sur le décideur pour ce qui est de l'optimisation du positionnement des produits et de l'information.

Les travaux en cours portent sur l'analyse et l'intégration de bases de données de tickets réels, ainsi que la construction d'un environnement de plus grande taille reproduisant les caractéristiques du magasin où ces tickets ont été collectés. Cela requiert notamment d'intégrer les positions effectives des articles et celles des panneaux de signalisation, le plan du magasin, etc. puisque nous avons montré que ces informations ont un impact direct sur les résultats de la simulation. Ces éléments doivent donc faire l'objet d'une validation auprès des experts commerciaux avec lesquels nous travaillons avant de pouvoir lancer des expérimentations grandeur nature.

6 Conclusion

La conception d'un outil intégré d'aide à la décision dans le domaine de la distribution est évidemment un objectif de longue haleine. Cependant nous avons montré comment combiner une approche de simulation incrémentale (adaptée à la représentation d'hypothèses portant sur les comportements individuels, les interactions entre agents ou la configuration de l'environnement) et des algorithmes d'exploration de données (employés d'ordinaire pour extraire un « comportement moyen » valable pour l'ensemble de la population). Notre proposition permet ainsi de doter les populations d'agents de profils différenciés, sur la base de critères statistiques réalistes, ces profils servant à leur tour à paramétrer les comportements des agents afin de produire des résultats similaires à ceux observés dans la réalité. En outre cette proximité des résultats de simulation avec les données enregistrées n'est pas que qualitative : les outils que nous avons construits en donnent une mesure. Par ailleurs, nous avons montré que ces algorithmes sont plutôt robustes au bruit dans les données. Il reste encore à tester cette approche sur une large échelle à partir d'une base de tickets réels.

Il est à noter que notre méthode, contrairement à la pratique courante en marketing, ne cherche pas à partitionner les transactions en « véritables » classes de clients, par exemple selon des critères socio-économiques, démographiques ou géographiques pré-établis. Nous

cherchons uniquement à capter des similarités dans les traces d'activité des individus, en construire une représentation abstraite et concise, dans l'hypothèse que cette représentation peut être génératrice de comportements. Cette démarche rend notre méthode indépendante de cadres théoriques préexistants, et surtout permet de changer aisément de point de vue. Ainsi par exemple, la prise en compte de variations saisonnières se réduit à utiliser alternativement les profils issus de données enregistrées à des périodes différentes.

Par ailleurs, nous comptons mettre à l'épreuve nos méthodes dans d'autres domaines d'application pour lesquelles nous avons de bonnes raisons de penser que nos techniques peuvent s'appliquer, c'est-à-dire ceux où les traces d'activité des agents s'expriment sous forme de transactions et leurs buts sous forme de prototypes. Cela devrait être notamment le cas en génomique fonctionnelle [2] (qui s'appuie fréquemment sur des approches de type classification multi-labels) ou en écologie [3] (où l'on rencontre de nombreuses méthodes d'identification de paramètres). Si la généralité de cette approche se confirme, elle constituera une étape importante pour le renforcement de l'intégration automatique de données dans les simulations muti-agents.

Références

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithm for mining association rules. In *Proceedings of the 20th Conference on Very Large Data Bases (VLDB'94)*, pages 487–499, 1994.

[2] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7) :830–836, 2006.

[3] Rémy Beaudouin, Gilles Monod, and Vincent Ginot. Selecting parameters for calibration via sensitivity analysis : An individual-based model of mosquitofish population dynamics. *Ecological Modelling*, 218 :29–48, 2008.

[4] Philippe Caillou and Javier Gil-Quijano. Description automatique de dynamiques de groupes dans des simulations à base d'agents. In *Actes des 20èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA'12)*, pages 23–32. Cépaduès, 2012.

[5] Luís Cavique. A scalable algorithm for the market basket analysis. *Journal of Retailing and Consumer Services*, 14(6), November 2007.

[6] Seung-Seok Choi, Sung-Hyuk Cha, and Charles C. Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1) :43–48, 2010.

[7] Chad Cumby, Andrew Fano, Rayid Ghani, and Marko Krema. Predicting customer shopping lists from point-of-sale purchase data. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data mining (KDD'04)*, pages 402–409, New York, NY, USA, 2004. ACM.

[8] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3) :297–302, 1945.

[9] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37 :547–579, 1901.

[10] Yoann Kubera, Philippe Mathieu, and Sébastien Picault. IODA : an interaction-oriented approach for multi-agent based simulations. *Journal of Autonomous Agents and Multi-Agent Systems*, pages 1–41, 2011.

[11] Benoît Lacroix, Philippe Mathieu, and Andras Kemeny. Formalizing the construction of populations in multi-agent simulations. *Journal of Engineering Applications of Artificial Intelligence*, 26(1) :211–226, January 2013.

[12] Peter Langfelder and Steve Horvath. Fast R functions for robust correlations and hierarchical clustering. *Journal of Statistical Software*, 46(11) :1–17, 2012.

[13] P. Mathieu, D. Panzoli, and S. Picault. Virtual customers in an agent world. In Y. Demazeau et al., editors, *Proceedings of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'12)*, Advances in Soft Computing. Springer, 2012.

[14] A. Ochiai. Zoogeographic studies on the soleoid fishes found in japan and its neighbouring regions. *Bulletin of the Japanese Society for Fish Science*, 22 :526–530, 1957.

[15] A. Schwaiger and B. Stahmer. Simmarket : Multiagent-based customer simulation and decision support for category management. In *Proceedings of MATES 2003 : multiagent system technologies*, volume 2831 of *LNAI*, pages 74–84. Springer, 2003.

[16] Pieter Sheth-Voss and Ismael E. Carreras. How informative is your segmentation ? a simple new metric yields surprising results. *Marketing Research*, pages 9–13, Winter 2010.

[17] Peer-Olaf Siebers, Uwe Aickelin, Helen Celia, and Chris W. Clegg. Using intelligent agents to understand management practices and retail productivity. In *Proceedings of the Winter Simulation Conference (WSC'07)*, pages 2212–2220, Washington, D.C., December 2007.

[18] Kubera Yoann, Philippe Mathieu, and Sébastien Picault. An interaction-oriented model of customer behavior for the simulation of supermarkets. In *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'10)*, pages 407–410. IEEE Computer Society, September 2010.

[19] T. Zhang and D. Zhang. Agent-based simulation of consumer purchase decision-making and the decoy effect. *Journal of Business Research*, 60(8) :912–922, August 2007. Complexities in Markets Special Issue.